



TITLE:

IDR定理をベースにした定常反復法の性能評価 (数値解析と数値計算アルゴリズムの最近の展開)

AUTHOR(S):

藤野, 清次; 尾上, 勇介; Sonneveld, P.; van Gijzen, M. B.

CITATION:

藤野, 清次 ...[et al]. IDR定理をベースにした定常反復法の性能評価 (数値解析と数値計算アルゴリズムの最近の展開). 数理解析研究所講究録 2010, 1719: 1-6

ISSUE DATE:

2010-11

URL:

<http://hdl.handle.net/2433/170368>

RIGHT:

IDR 定理をベースにした定常反復法の性能評価

A performance estimation of stationary iterative methods based on IDR Theorem

藤野清次 (九州大学) 尾上勇介 (九州大学大学院)

Fujino, S., Onoue, Y., Kyushu University

Sonneveld, P. and van Gijzen, M.B. (Delft University of Technology)

Abstract: The conventional SOR method is well known to be a simple one of stationary iterative methods for solving a linear system of equations with nonsymmetric coefficient matrix, but it converges slowly and sometimes stagnates during iterations. Therefore, we improve the SOR method by means of the extended Induced Dimension Reduction (IDR) Theorem proposed by Sonneveld *et al.* in 2008 in order to gain robustness of convergence. In this article, we devise the IDR-based SOR method with parameter s . A number of numerical experiments verify effectiveness and robustness of the IDR-based SOR method. Characteristics of convergence of IDR-based SOR method will be effective for a rich variety of applications.

1. はじめに

最近, 拡張 IDR(Induced Dimension Reduction) 定理に基づく反復解法が続々と誕生している [3][6-10][12-13][15-17][19][21]. 本稿では, 代表的な定常反復法の一つである SOR(Successive Over-Relaxation) 法に定理を適用し, IDR based SOR(以下, IDR(s)-SOR と呼ぶ) 法を提案し, その性能評価と比較を行う. そして, 数値実験により, 従来の反復法に比べて, IDR(s)-SOR 法の収束優位性を立証するものである [2][4][5][14].

1 IDR-SOR 法の算法

IDR-SOR 法の算法を以下に示す. ここでは, 任意ベクトル \mathbf{p} には乗算合同法による一様乱数を与えた.

算法 1: IDR-SOR 法

1. Let \mathbf{x}_0 be an initial solution,
2. put $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$,
3. let \mathbf{p} be a random vector,
4. set $\gamma = 0$,
5. for $n = 0, 1, \dots$,
6. $\mathbf{s}_n = (L + D/\omega)^{-1}(\mathbf{r}_n - \gamma d\mathbf{r}_n)$,
7. $d\mathbf{x}_{n+1} = \mathbf{s}_n - \gamma d\mathbf{x}_n$,
8. $d\mathbf{r}_{n+1} = -((1 - 1/\omega)D + U)\mathbf{s}_n - \mathbf{r}_n$,
9. $\mathbf{r}_{n+1} = \mathbf{r}_n + d\mathbf{r}_{n+1}$, $\mathbf{x}_{n+1} = \mathbf{x}_n + d\mathbf{x}_{n+1}$,
10. if $\|\mathbf{r}_{n+1}\|_2/\|\mathbf{r}_0\|_2 \leq \epsilon$ then stop,
11. $\gamma = (\mathbf{p}, \mathbf{r}_{n+1})/(\mathbf{p}, d\mathbf{r}_{n+1})$,
13. end for,

2 IDR(s)-SOR 法の算法

IDR(s)-SOR 法の算法を以下に示す. ここで, s 次元ベクトル \mathbf{e}_k は, k 番目の要素が 1 で, それ以外の要素が 0 とする.

算法 2: IDR(s)-SOR 法

1. Let \mathbf{x}_0 be an initial solution,
2. put $\mathbf{r}_0 = \mathbf{b} - A\mathbf{x}_0$,
3. $P = (\mathbf{p}_1 \mathbf{p}_2 \dots \mathbf{p}_s) \in R^{N \times s}$, set $\gamma = 0$,
{initial loop: build matrices $E = (d\mathbf{r}_1 d\mathbf{r}_2 \dots d\mathbf{r}_s)$,
 $Q = (d\mathbf{x}_1 d\mathbf{x}_2 \dots d\mathbf{x}_s)$ by ISOR method}}
5. for $n = 0, 1, \dots, s-1$
6. $\mathbf{s}_n = (L + \frac{1}{\omega}D)^{-1}(\mathbf{r}_n - \gamma d\mathbf{r}_n)$,
7. $d\mathbf{x}_{n+1} = \mathbf{s}_n - \gamma d\mathbf{x}_n$,
8. $d\mathbf{r}_{n+1} = -((1 - \omega^{-1})D + U)\mathbf{s}_n - \mathbf{r}_n$,
9. $\mathbf{r}_{n+1} = \mathbf{r}_n + d\mathbf{r}_{n+1}$, $\mathbf{x}_{n+1} = \mathbf{x}_n + d\mathbf{x}_{n+1}$,
10. if $\|\mathbf{r}_{n+1}\|_2/\|\mathbf{r}_0\|_2 \leq \epsilon$ then stop,
11. $\gamma = (\mathbf{p}_1, \mathbf{r}_{n+1})/(\mathbf{p}_1, d\mathbf{r}_{n+1})$,
12. $E\mathbf{e}_{n+1} = d\mathbf{r}_{n+1}$, $Q\mathbf{e}_{n+1} = d\mathbf{x}_{n+1}$,
13. end for,
14. $M = P^T E$, $\mathbf{f} = P^T \mathbf{r}_s$,
15. $n = s$, $k = 1$,
{main loop}
16. while $\|\mathbf{r}_{n+1}\|_2/\|\mathbf{r}_0\|_2 > \epsilon$
17. solve \mathbf{c} from $M\mathbf{c} = \mathbf{f}$,
18. $\mathbf{s}_n = (L + D/\omega)^{-1}(\mathbf{r}_n - E\mathbf{c})$,
19. $d\mathbf{x}_{n+1} = \mathbf{s}_n - Q\mathbf{c}$,
20. $d\mathbf{r}_{n+1} = -((1 - \omega^{-1})D + U)\mathbf{s}_n - \mathbf{r}_n$,
21. $\mathbf{r}_{n+1} = \mathbf{r}_n + d\mathbf{r}_{n+1}$, $\mathbf{x}_{n+1} = \mathbf{x}_n + d\mathbf{x}_{n+1}$,

22. $Ee_k = dr_{n+1}, \quad Qe_k = dx_{n+1},$
 23. $Me_k = P^T dr_{n+1}, \quad f = f + Me_k,$
 24. $n = n + 1, \quad k = k + 1,$
 25. if $k > s$ then $k = 1,$
 26. end while.

2.1 SOR 法から IDR-SOR 法へ

以下の手順で, SOR 法から IDR-SOR 法が導出される.

- $r_{k+1} = Br_k \implies r_{k+1} = B(r_k + \gamma_k dr_k).$
 ここで, $dr_k = r_k - r_{k-1}$: 残差ベクトルの差, γ_k はスカラー値を表す.
- パラメータ γ_k は任意のベクトル p と $(r_k + \gamma_k dr_k)$ が直交するように決定される. すなわち, 内積について $(p, r_k + \gamma_k dr_k) = 0$ の関係から γ_k は求められる.
- 任意のベクトル p には初期残差ベクトル r_0 が代入される.

2.2 IDR-SOR 法から IDR(s)-SOR 法へ

以下の手順で, IDR-SOR 法から IDR(s)-SOR 法が導出される.

- $r_{k+1} = B(r_k + \gamma_k dr_k) \implies r_{k+1} = B(r_k + \sum_{j=0}^s \gamma_{k-j} dr_{k-j}).$ ここで, γ_{k-j} はスカラー値とする.
- γ_{k-j} は, ベクトル $v_k (= (r_k + \sum_{j=0}^s \gamma_{k-j} dr_{k-j}))$ が P^T の零空間となるように決められる. すなわち, $N \times s$ 行列 $P = (p_1 \ p_2 \ \dots \ p_s)$ の転置 P^T がベクトル v_k と直交する. 具体的には, 内積 $P^T v_k = 0$ により決定される.
- 任意ベクトル $p_j, (j = 0, \dots, s)$ には, 初期残差ベクトル r_0 または乱数が代入され, Gram-Schmidt の直交化が施される.

3. 数値実験

3.1 テスト問題

Table 1 に 24 個のテスト行列の主な特徴を以下に示す. 行列 waseda, w_dense は, 早稲田大学若尾研提供の行列で, その他 22 行列は Florida 大学疎行列 DB[18] から選出した.

Table 1: テスト行列の主な特徴.

行列	次元数	総非零要素	平均非零要素	$\log_{10}(\text{domi.})$
airfoil_2D	14,214	259,688	18.270	0.49
big	13,209	91,465	6.924	0.71
dc2	116,835	766,396	6.560	0.01
dc3	116,835	766,396	6.560	0.01
ecl32	51,993	380,415	7.317	-1.87
epb1	14,734	95,053	6.451	0.73
epb2	25,228	175,027	6.938	0.74
epb3	84,617	463,625	5.479	0.65
k3plates	11,107	378,927	34.116	1.11
language	399,130	1,216,334	3.047	0.30
memplus	17,758	99,147	5.583	1.04
poisson3Da	13,514	352,762	26.103	1.20
poisson3Db	85,623	2,374,949	27.737	1.19
raefsky2	3,242	293,551	90.546	1.42
raefsky3	21,200	1,488,768	70.225	1.22
sme3Da	12,504	874,887	69.969	1.27
sme3Db	29,067	2,081,063	71.595	1.28
sme3Dc	42,930	3,148,656	73.344	1.29
trans4	116,835	766,396	6.560	1.06
wang3	26,064	177,168	6.797	0.77
wang4	26,068	177,196	6.797	0.75
waseda	19,060	24,377,548	1670.8	1.72
w_dense	7,601	57,775,201	7601.0	1.57
xenon1	48,600	1,181,120	24.303	0.97

各行列の対角優位度の大きさを表す指標として $\log_{10}(\text{domi.})$ が考えられる. その定義を以下に示す.

$$\text{domi.} = \frac{\sum_i |a_{i,i}|/nd}{\sum_{i \neq j} |a_{i,j}|/nnd}. \quad (1)$$

ここで, “nd” は対角要素数, “nnd” は非対角の要素数を各々表す. ただ, 今回の数値実験では, IDR(s)-SOR 法の収束性とこの指標との相関はとれなかった.

3.2 計算機環境と計算条件

計算機環境と計算条件は, 以下の通りである.

計算機環境

1. 計算機は, HP Workstation xw4200(CPU: Intel(R) Pentium(R) 4, クロック周波数: 3.8GHz, メモリ: 3Gbyte, OS: Suse Linux version 9.2).
2. プログラムは Fortran90 で, コンパイラは Intel Compiler version 10.1 を用いた.
3. 最適化オプションは “-O3” を使用した.
4. 計算はすべて倍精度浮動小数点演算で行った.

計算条件

1. 収束判定値は相対残差の 2 ノルム：
 $\|r_{n+1}\|_2/\|r_0\|_2 \leq 10^{-6}$ である。
2. 疎行列ベクトル積の打ち切り回数は 2 万回とした。
3. 右辺項には物理的条件から得られる値を用いた。
4. 対角 scaling により対角項を 1 に正規化した。
5. ILU(0) 分解つき GMRES(k) 法 [11], 同 BiCGStab 法 [20], 同 GPBi-CG 法 [22], 同 BiCGSafe 法 [1], 同 Bi-IDR 法 [19], 前処理なし IDR(s)-SOR 法の収束性を調べた。
6. IDR(s)-SOR 法の加速係数 ω は区間 [1.0, 1.90] で 0.1 刻みで 10 通り調べた。
7. IDR(s)-SOR 法, Bi-IDR(s) 法の s の値は, 1, 2, 4, 8 の計 4 通り調べた。
8. BiCGStab 法, GPBi-CG 法, BiCGSafe 法の初期シャドウ残差 r_0^* は初期残差 r_0 を用いた。
9. GMRES(k) 法のリスタート係数 k は, 10, 20, 50, 100, 200, 500, 1000, 2000, 5000 の計 9 通り調べた。

2.3 実験結果 1

Table 2 に, IDR(s)-SOR, Bi-IDR(s) 法の収束までに要した行列ベクトル積の回数を表す。同様に, Table 3. に, GMRES(k) 法の収束までに要した行列ベクトル積の回数を表す。表中の “max” は, 疎行列ベクトル積の打ち切り回数の 2 万回までに収束しなかったことを示す。SOR 法の加速係数 ω の値は最適値を表す。この表から以下の観察ができる。

- 例えば, IDR(s)-SOR 法について, 行列 airfoil_2d では, $s=1$ のとき 259 回が $s=8$ のとき 238 回 (約 92%) に減ったことが分かる。しかし, 後述の Table 6 からわかるように, $s=1$ のとき 0.45 秒が $s=8$ のとき 0.55 秒 (約 122%) に逆に増加したことがわかる。
- Bi-IDR(s) 法についても, 同様の傾向が見られる。すなわち, 行列 airfoil_2d では, $s=1$ のとき 387 回が $s=8$ のとき 331 回 (約 86%) に減ったことが分かる。しかし, $s=1$ のとき 0.57 秒が $s=8$ のとき 0.80 秒 (約 140%) に逆に大幅に増加した (表は割愛)。
- このように, IDR(s)-SOR 法や Bi-IDR(s) 法の系統の反復法では, 疎行列ベクトル積の回数と収束までの計

算時間との間は比例関係ではなく, 疎行列ベクトル積の回数の “多いか少ないか” は反復法の性能を比較する上で指標になり難いことがわかった。

- 一方, GMRES(k) 法では, 行列 airfoil_2d において, $k=20$ のとき 1234 回が $k=1000$ のとき 288 回 (約 23.3%) に大幅に減ったことが分かる。計算時間については, $k=20$ のとき 2.28 秒が $k=1000$ のとき 2.01 秒 (約 88.2%) 少し減少した (表は割愛)。ここでも, 疎行列ベクトル積の回数の “多いか少ないか” は, 性能評価の判断材料の目安にし難い。

Table 2: IDR(s)-SOR, Bi-IDR(s) 法の行列ベクトル積回数。

matrix	ω	IDR(s)-SOR				Bi-IDR(s)			
		$s=1$	$s=2$	$s=4$	$s=8$	$s=1$	$s=2$	$s=4$	$s=8$
airfoil_2d	1.9	259	250	248	238	387	359	332	331
big	1.9	684	428	373	361	4719	1977	1138	856
dc2	1.9	184	149	129	122	439	256	218	231
dc3	1.2	1048	576	275	164	2571	820	456	439
ecl32	1.8	293	245	231	216	441	459	358	353
epb1	1.5	472	349	320	292	511	505	455	443
epb2	1.2	152	129	128	125	230	215	216	210
epb3	1.0	1372	1057	932	843	2419	1900	1668	1544
k3plates	1.2	8396	max	2047	3354	344	375	371	508
language	1.0	16	15	15	14	43	41	36	32
memplus	1.9	311	197	190	144	243	149	150	123
poi-3Da	1.8	79	78	83	77	146	148	134	131
poi-3Db	1.8	224	214	222	211	348	320	315	291
raefsky2	1.0	350	245	217	196	609	448	394	354
raefsky3	1.0	1765	1653	1340	1251	2673	2863	2322	2243
sme3Da	1.9	5239	1236	1072	992	5149	4024	2543	2239
sme3Db	1.9	9640	1503	1219	1136	7147	4946	3272	2681
sme3Dc	1.9	max	3263	1626	1484	max	max	4261	3722
trans4	1.6	124	113	98	95	477	214	208	216
wang3	1.7	130	122	115	114	209	200	184	182
wang4	1.8	134	135	131	121	213	173	184	158
waseda	1.0	165	136	135	147	878	538	408	694
w.dense	1.0	63	46	35	32	46	43	41	40
xenon1	1.8	1350	1029	901	894	924	775	717	805

Table 3: GMRES(k) 法の行列ベクトル積の回数。

matrix	parameter: k								
	10	20	50	100	200	500	10^3	2000	5000
airfoil_2d	max	1234	580	378	349	288	288	288	288
big	max	max	max	max	6547	528	506	506	506
dc2	471	309	167	142	133	133	133	133	133
dc3	max	6557	2484	563	337	264	264	264	264
ecl32	5646	4876	3137	2186	792	315	315	315	315
epb1	1212	880	592	492	511	398	398	398	398
epb2	398	355	257	215	188	188	188	188	188
epb3	max	max	max	5425	3419	2357	1640	1337	1337
k3plates	1502	799	401	300	291	264	264	264	264
language	97	40	30	30	30	30	30	30	30
memplus	122	87	74	69	69	69	69	69	69
poi-3Da	184	175	140	130	113	113	113	113	113
poi-3Db	809	703	401	299	249	237	237	237	237
raefsky2	3841	2475	2138	977	604	316	316	316	316
raefsky3	3723	3441	3020	2837	2417	1940	1780	1548	1548
sme3Da	max	max	max	max	max	max	max	1557	1557
sme3Db	max	max	max	max	max	max	max	2027	2022
sme3Dc	max	max	max	max	max	max	max	9279	2782
trans4	489	391	197	164	154	154	154	154	154
wang3	532	375	236	218	160	160	160	160	160
wang4	max	max	409	184	141	141	141	141	141
waseda	267	114	64	53	53	53	53	53	53
w.dense	56	40	34	34	34	34	34	34	34
xenon1	8807	4398	1880	1098	747	672	614	614	614

- 上で見たように、同じ反復法でも、疎行列ベクトル積の回数と収束までの計算時間との相関を取れない。したがって、異なる反復法どうしの収束性比較を疎行列ベクトル積の回数で見極めは再考の余地がある。
- このような状況が起こるのは、算法中に占める内積計算の割合が大きく異なる、ことに起因すると思われる。

次に、Table 4 に IDR(s)-SOR 法と Bi-IDR(s) 法のメモリ使用量 [単位: MBytes] を示す。同様に、Table 5 に GMRES(k) 法のメモリ使用量を示す。各解法の最適なパラメータのときのメモリ使用量を太字で示す。GMRES(k) 法のメモリ使用量が IDR(s)-SOR 法と Bi-IDR(s) 法のそれらに比べて非常に多いことがわかる。

Table 4: IDR(s)-SOR 法と Bi-IDR(s) 法のメモリ使用量。

matrix	IDR(s)-SOR				Bi-IDR(s)			
	$s=1$	$s=2$	$s=4$	$s=8$	$s=1$	$s=2$	$s=4$	$s=8$
airfoil_2d	7.68	8.00	8.66	9.96	4.11	4.44	5.09	6.39
big	3.71	4.01	4.61	5.82	2.11	2.41	3.01	4.22
dc2	31.80	34.48	39.83	50.52	18.13	20.80	26.15	36.85
dc3	31.80	34.48	39.83	50.52	18.13	20.80	26.15	36.85
ec132	15.05	16.24	18.62	23.39	8.52	9.71	12.09	16.85
epb1	3.97	4.31	4.99	6.34	2.27	2.61	3.28	4.63
epb2	7.09	7.66	8.82	11.13	4.02	4.60	5.76	8.07
epb3	20.94	22.88	26.75	34.50	12.08	14.02	17.90	25.64
k3plates	10.03	10.28	10.79	11.81	5.23	5.48	5.99	7.01
language	76.56	85.70	103.	140.	45.89	55.03	73.30	109.
memplus	5.06	5.46	6.28	7.90	2.87	3.27	4.09	5.71
poi-3Da	9.72	10.03	10.65	11.89	5.12	5.43	6.05	7.29
poi-3Db	64.81	66.77	70.69	78.53	34.04	36.00	39.92	47.76
raefsky2	7.13	7.21	7.35	7.65	3.63	3.70	3.85	4.15
raefsky3	36.66	37.15	38.12	40.06	18.74	19.22	20.19	22.13
sme3Da	21.55	21.84	22.41	23.56	11.01	11.30	11.87	13.02
sme3Db	51.18	51.85	53.18	55.84	26.14	26.81	28.14	30.80
sme3Dc	77.31	78.29	80.26	84.19	39.47	40.46	42.42	46.35
trans4	31.80	34.48	39.83	50.52	18.13	20.80	26.15	36.85
wang3	7.24	7.83	9.03	11.41	4.12	4.71	5.91	8.29
wang4	7.24	7.84	9.03	11.42	4.12	4.71	5.91	8.29
waseda	560	560	561	563	280	280	281	283
w_dense	1323	1323	1323	1324	661	661	662	663
xenon1	32.97	34.08	36.30	40.75	17.41	18.52	20.75	25.20

Table 5: GMRES(k) 法のメモリ使用量。

matrix	parameter: k						
	20	50	100	200	500	10 ³	2000
airfoil_2d	5.96	9.23	14.71	25.79	59.93	119.	251.
big	3.82	6.86	11.96	22.27	54.11	110.	233.
dc2	33.29	60.05	104.	194.	463.	914.	1828.
dc3	33.29	60.05	104.	194.	463.	914.	1828.
ec132	15.27	27.18	47.08	86.98	207.	411.	831.
epb1	4.18	7.57	13.25	24.72	60.06	122.	257.
epb2	7.30	13.09	22.77	42.25	101.	203.	418.
epb3	23.06	42.45	74.79	139.	334.	663.	1331.
k3plates	6.67	9.23	13.53	22.23	49.26	97.	205.
language	97.66	189.	341.	646.	1561.	3089.	6157.
memplus	5.17	9.26	16.09	29.87	72.12	145.	304.
poi-3Da	6.88	9.99	15.20	25.74	58.29	115.	241.
poi-3Db	45.15	64.76	97.48	163.	360.	692.	1369.
raefsky2	4.05	4.81	6.11	8.81	17.84	35.	83.
raefsky3	21.49	26.36	34.50	50.91	101.	187.	372.
sme3Da	12.64	15.52	20.35	30.12	60.35	113.	232.
sme3Db	29.92	36.59	47.74	70.14	138.	254.	499.
sme3Dc	45.04	54.89	71.32	104.	204.	373.	724.
trans4	33.29	60.05	104.	194.	463.	914.	1828.
wang3	7.50	13.48	23.48	43.	104.	210.	431.
wang4	7.50	13.48	23.49	43.61	104.	210.	431.
waseda	282.	287.	294.	309.	354.	433.	601.
w_dense	662.	664.	667.	673.	692.	727.	808.
xenon1	23.72	34.86	53.46	90.77	203.	394.	788.

2.4 実験結果 2

Table 6: 最適加速係数 ω と IDR(s)-SOR 法の計算時間。

matrix	ω	IDR(s)-SOR			
		$s=1$	$s=2$	$s=4$	$s=8$
airfoil_2d	1.9	0.45	0.45	0.50	0.55
big	1.9	0.72	0.49	0.50	0.59
dc2	1.8	2.39	2.19	2.34	3.14
dc3	1.2	13.92	8.37	4.96	4.14
ec132	1.8	1.58	1.48	1.71	2.14
epb1	1.5	0.49	0.40	0.43	0.50
epb2	1.2	0.32	0.29	0.32	0.38
epb3	1.0	11.79	10.32	11.21	14.16
k3plates	1.2	15.09	∞	3.95	7.24
language	1.0	0.81	0.86	1.02	1.46
memplus	1.9	0.42	0.30	0.33	0.31
poi-3Da	1.8	0.20	0.20	0.22	0.23
poi-3Db	1.8	4.63	4.72	5.46	6.26
raefsky2	1.0	0.41	0.29	0.27	0.26
raefsky3	1.0	10.33	10.09	8.51	8.40
sme3Da	1.9	21.98	5.29	4.77	4.68
sme3Db	1.9	103.2	16.24	13.93	13.69
sme3Dc	1.9	∞	58.40	30.12	30.81
trans4	1.6	1.63	1.84	1.76	2.36
wang3	1.7	0.28	0.29	0.30	0.36
wang4	1.8	0.29	0.31	0.34	0.37
waseda	1.0	13.81	11.40	11.29	12.29
w_dense	1.0	13.51	10.80	8.22	7.54
xenon1	1.8	10.04	8.34	8.34	10.21

Table 7: ILU(0) 前処理つき BiCGStab 法, GPBi-CG 法, BiCGSafe 法の計算時間。

matrix	前処理なし			ILU(0)+		
	BiCG-Stab	GP-BiCG	BiCG-Safe	BiCG-Stab	GP-BiCG	BiCG-Safe
airfoil_2d	0.52	∞	0.61	0.25	0.24	0.23
big	3.14	2.12	1.96	1.87	2.34	1.73
dc2	3.88	3.54	3.50	2.17	2.30	2.19
dc3	20.80	17.86	15.83	5.66	8.11	7.34
ec132	1.85	2.65	2.39	1.07	1.24	1.16
epb1	0.32	0.43	0.41	0.16	0.18	0.17
epb2	0.27	0.41	0.37	0.10	0.11	0.11
epb3	10.13	11.55	11.03	0.83	1.00	0.99
k3plates	0.58	0.56	0.54	3.04	4.25	4.98
language	1.11	1.42	2.92	0.92	0.97	0.96
memplus	0.18	0.14	0.13	0.31	0.35	0.33
poi-3Da	0.34	0.27	0.28	0.37	0.32	0.31
poi-3Db	8.65	7.85	7.97	6.18	5.69	5.23
raefsky2	0.60	0.56	0.55	0.20	0.20	0.20
raefsky3	17.15	17.91	17.01	1.70	1.74	1.75
sme3Da	23.94	18.79	14.68	15.52	11.18	10.43
sme3Db	65.41	64.50	58.90	31.25	29.75	25.70
sme3Dc	287.82	318.96	217.67	65.10	75.78	70.89
trans4	3.26	3.05	3.90	1.11	1.19	1.21
wang3	0.33	0.36	0.36	0.25	0.24	0.21
wang4	0.25	0.35	0.37	0.17	0.20	0.18
waseda	42.77	34.88	25.40	181.10	177.21	177.89
w_dense	11.28	9.32	9.91	671.39	667.15	665.88
xenon1	5.19	6.05	4.48	∞	∞	∞

ここでは、IDR(s)-SOR 法の計算時間とフィルインを考慮しない ILU(0) 分解付きの従来の反復法との計算時間 [単位: 秒] の比較を行う。Table 6 に最適加速係数 ω とそのときの

ISOR(s) 法の計算時間を示す。表中の“ ∞ ”記号は収束しなかったことを表す。Table 7 に ILU(0) 前処理つき BiCGStab 法, GPBi-CG 法, BiCGSafe 法の計算時間を示す。

Table 8: ILU(0) 前処理つき GMRES(k) 法の計算時間。

matrix	parameter : k					
	20	50	100	200	500	10^3
airfoil_2d	0.26	0.23	0.24	0.23	0.23	0.23
big	∞	11.3	3.40	1.37	1.21	1.24
dc2	3.63	4.44	5.70	5.93	5.89	5.92
dc3	21.2	8.48	8.54	10.0	10.0	10.0
ecl32	8.36	4.13	3.14	1.99	1.96	1.91
epb1	0.22	0.25	0.28	0.27	0.27	0.27
epb2	0.12	0.12	0.12	0.12	0.12	0.12
epb3	1.64	2.37	1.89	1.96	2.19	2.11
k3plates	∞	2.68	1.34	1.40	1.30	1.28
language	0.97	0.98	0.97	0.98	0.98	0.97
memplus	0.41	0.38	0.43	0.43	0.43	0.43
poi-3Da	0.35	0.33	0.33	0.34	0.34	0.34
poi-3Db	7.86	6.86	6.84	6.78	6.74	6.89
raefsky2	0.26	0.16	0.16	0.16	0.17	0.16
raefsky3	1.23	0.93	0.93	0.93	0.93	0.93
sme3Da	∞	∞	∞	74.1	6.10	6.30
sme3Db	∞	∞	∞	∞	53.2	25.2
sme3Dc	∞	∞	∞	271.	246.	64.6
trans4	1.74	2.31	2.62	2.61	2.59	2.63
wang3	0.28	0.26	0.26	0.28	0.26	0.26
wang4	0.29	0.23	0.23	0.23	0.23	0.23
waseda	175.	173.	173.	173.	173.	172.
w_dense	664.	666.	666.	665.	666.	665.
xenon1	∞	∞	∞	∞	197.	89.3

Table 9: ILU(0) 前処理つき Bi-ILDR(s) 法の計算時間。

matrix	ILU(0) + Bi-ILDR(s)			
	$s = 1$	$s = 2$	$s = 4$	$s = 8$
airfoil_2d	0.23	0.21	0.22	0.27
big	2.18	1.03	0.68	0.71
dc2	2.41	2.25	2.49	3.08
dc3	6.77	5.71	4.34	5.19
ecl32	1.47	1.16	1.27	1.46
epb1	0.16	0.16	0.18	0.21
epb2	0.11	0.11	0.13	0.16
epb3	0.86	0.97	0.98	1.47
k3plates	3.43	1.84	1.51	1.36
language	0.93	0.96	1.14	1.44
memplus	0.31	0.31	0.31	0.36
poi-3Da	0.33	0.34	0.34	0.34
poi-3Db	5.15	5.58	5.49	5.50
raefsky2	0.20	0.18	0.18	0.17
raefsky3	1.63	1.38	1.28	1.19
sme3Da	10.89	5.76	4.68	4.40
sme3Db	28.55	19.37	16.28	14.32
sme3Dc	79.96	50.13	35.85	32.02
trans4	1.10	1.31	1.40	1.81
wang3	0.22	0.21	0.22	0.28
wang4	0.20	0.19	0.22	0.28
waseda	178.	179.	178.	178.
w_dense	674.	673.	674.	672.
xenon1	∞	∞	∞	∞

Table 8 に ILU(0) 分解つき GMRES(k) 法の計算時間を表す。Table 9 に ILU(0) 分解つき Bi-ILDR(s) 法の計算時間を表す。

Table 10 に IDR(s)-SOR および ILU(0) 分解 Bi-ILDR(s), GMRES(k), BiCGStab 法, BiCGSafe 法の最適パラメータのときの計算時間を示す。また、括弧内の数字は最適パラメータの値を示す。ただし、GPBi-CG 法の結果は割愛した。太字の数字は、各行列で最も収束までの計算時間が少なかったケースを表す。また、下線を付けた数字は、IDR(s)-SOR 法が他の反復法に比べて非常に遅かったケースを表す。

Table 10: IDR(s)-SOR, ILU(0) 分解 Bi-ILDR(s), 同 GMRES(k), 同 BiCGStab 法, 同 BiCGSafe 法の計算時間。

matrix	IDR(s)-SOR (s, ω)	ILU(0) +			
		Bi-ILDR(s)	GMRES(k)	BiCG-Stab	BiCG-Safe
airfoil_2d	0.45 (1, 1.9)	0.21 ($s=2$)	0.23 ($k=50$)	0.25	0.23
big	0.49 (2, 1.9)	0.68 ($s=4$)	1.21 ($k=500$)	1.87	1.73
dc2	2.19 (2, 1.8)	2.25 ($s=2$)	3.63 ($k=20$)	2.17	2.19
dc3	4.14 (8, 1.2)	4.34 ($s=4$)	8.48 ($k=50$)	5.66	7.34
ecl32	1.48 (2, 1.8)	1.16 ($s=2$)	1.91 ($k=10^3$)	1.07	1.16
epb1	0.40 (2, 1.5)	0.16 ($s=1$)	0.21 ($k=10$)	0.16	0.17
epb2	0.29 (2, 1.2)	0.11 ($s=1$)	0.12 ($k=10$)	0.10	0.11
epb3	10.3 (2, 1.0)	0.86 ($s=1$)	1.64 ($k=20$)	0.83	0.99
k3plates	3.95 (4, 1.2)	1.36 ($s=8$)	1.28 ($k=10^3$)	3.04	4.98
language	0.81 (1, 1.0)	0.93 ($s=1$)	0.97 ($k=10$)	0.92	0.96
memplus	0.30 (2, 1.9)	0.31 ($s=1$)	0.38 ($k=50$)	0.31	0.33
poi-3Da	0.20 (1, 1.8)	0.33 ($s=1$)	0.33 ($k=50$)	0.37	0.31
poi-3Db	4.63 (1, 1.8)	5.15 ($s=1$)	6.78 ($k=200$)	6.18	5.23
raefsky2	0.26 (8, 1.0)	0.17 ($s=8$)	0.16 ($k=50$)	0.20	0.20
raefsky3	<u>8.40</u> (8, 1.0)	1.19 ($s=8$)	0.93 ($k=50$)	1.70	1.75
sme3Da	4.68 (8, 1.9)	4.40 ($s=8$)	6.10 ($k=500$)	15.52	10.43
sme3Db	13.69 (8, 1.9)	14.32 ($s=8$)	25.2 ($k=10^3$)	31.25	25.70
sme3Dc	30.81 (8, 1.9)	32.02 ($s=8$)	64.6 ($k=10^3$)	65.10	70.89
trans4	1.63 (1, 1.6)	1.10 ($s=1$)	1.72 ($k=10$)	1.11	1.21
wang3	0.28 (1, 1.7)	0.21 ($s=2$)	0.26 ($k=50$)	0.25	0.21
wang4	0.29 (1, 1.8)	0.19 ($s=2$)	0.23 ($k=50$)	0.17	0.18
waseda	11.29 (4, 1.0)	178. ($s=1$)	172. ($k=10^3$)	181.	177.
w_dense	7.54 (8, 1.0)	672. ($s=8$)	664. ($k=20$)	671.	665.
xenon1	8.34 (2, 1.8)	∞	89.3 ($k=10^3$)	∞	∞

Table 11 に IDR(s)-SOR 法の収束性比較をまとめた。比較する反復法の対応は次のように行った。

1. IDR(s)-SOR 法の加速係数 $\omega = 1$ に固定の場合：従来の前処理なし反復法
2. IDR(s)-SOR 法の加速係数可変 ($1.0 \leq \omega \leq 1.9$) の場合：従来の ILU(0) 分解前処理つき反復法

この表から、IDR(s)-SOR 法の収束性のよさがわかる。ただし、最少時間が同じ (計測最小単位: 0.01 秒) 場合は、重複してカウントをした。そのため、行列の個数 24 個よりも合計行列数は多くなった。Bi-ILDR(s) 法の性能も非常によい。また、BiCGStab 法の健闘も目立つ。

Table 11: IDR(s)-SOR 法の収束性比較のまとめ。

method	parameter ω	
	$\omega = 1$: 固定	ω : 最適
IDR(s)-SOR	10	11
	前処理なし	ILU(0) 分解
Bi-ILDR(s)	6	6
BiCGStab	4	5
GPBi-CG	0	0
BiCGSafe	3	1
GMRES(k)	2	2
Total counts	25	25

Figure 1(a),(b) に、加速係数 ω を変動させたときの行列 epb3 と sme3Da における IDR(s)-SOR 法の反復回数の変化の様子を表す。パラメータ s は 1, 2, 4, 8 の 4 つの場合である。行列 epb3 では、加速係数 ω が 1 を越えると収束しなくなった。一方、行列 sme3Da では、加速係数 ω が 1.9 まで収束しかつ反復回数が最も少なかった。

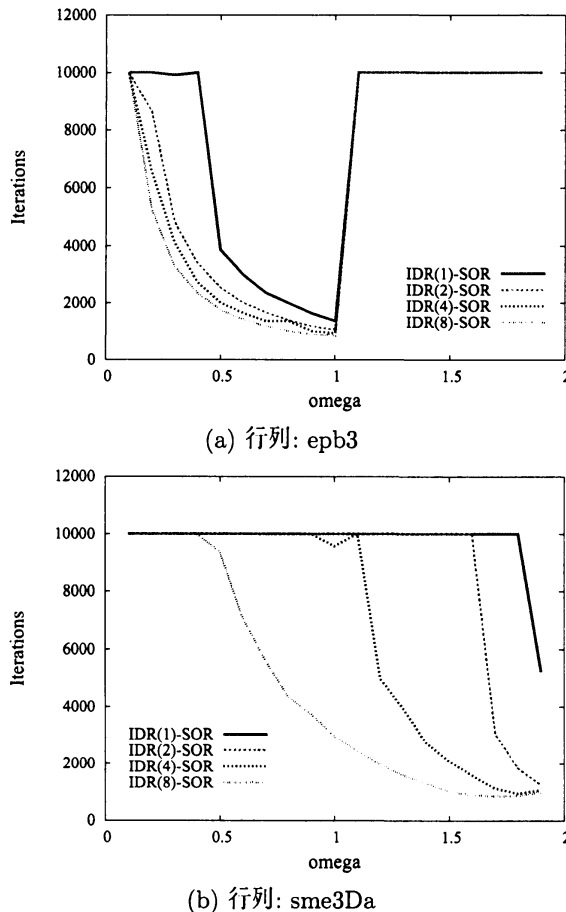


Figure 1: IDR(s)-SOR 法の反復回数の変化の様子。

参考文献

- [1] 藤野清次, 藤原牧, 吉田正浩: 準残差の最小化に基づく BiCGSafe 法の収束性について, Trans. of JSCES, Paper No.20050028, 2005.
- [2] Fujino, S., Sonneveld, P. and van Gijzen, M.B. and Onoue, Y., Application of the IDR Theorem to Stationary Iterative Methods and their Performance Evaluation, The Abstract of SIAM LA09, Monterey, 26th-29th, Oct., 2009.
- [3] Gutknecht, M., IDR explained, Dec. 2008, Oct., 2009, to appear in Electr. Trans. Numer. Anal.
- [4] Harumatsu, M., Kusakabe, Y., Fujino, S., Fukushige, T., Arima, T. and Sonneveld, P., A Proposal of Gauss-Seidel and Successive Over-Relaxation Methods based on IDR Theorem, Technical report of Information Processing Society of Japan, JAXA Chofu, June, 2009. (In Japanese)
- [5] Kusakabe, Y., Fujino, S., A proposal of Jacobi method based on extended Induced Dimension Reduction Theorem aimed at high convergence rate, The Proc. of EMAC2009, Adelaide, December, 2009. (To appear)
- [6] Nakashima, N., Fujino, S., Tateiba, M. and Onoue, Y., A State-of-the-Art Linear Solver IDR(s) Method for Large Scale Electromagnetic Multiple Scattering Simulations, Proc. the 2009 IEEE International Symposium on Antennas and Propagation and USNC-URSI National Radio Science Meeting, CD-ROM, 2009.6.
- [7] Nakashima, N., Fujino, S. and Tateiba, M., A state-of-the-art linear solver IDR(s) method for large scale electromagnetic multiple scattering simulations, ACES2010, Tampere, Finland, 2010. (To appear)
- [8] 尾上 勇介, 藤野 清次, 中嶋 徳正, IDR(s) 法の簡便な前処理と重厚な前処理の違いについて, Transactions of JSCES, Vol. 2008, 20080023, 2008 年 9 月.
- [9] 尾上 勇介, 藤野 清次, IDR(s) 法系統の反復法に適用可能な計算量削減の工夫, 日本応用数理学会論文誌, **19**, 3(2009), pp.329-350.
- [10] 尾上 勇介, 藤野 清次, BiCGStab(s, L) 法の収束安定性の向上, 投稿中.
- [11] Saad, Y., Schultz, M. H., GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM J. Sci. Stat. Comput., **7**, 3, pp.856-869, 1986.
- [12] Simoncini, V., Szyld, D., Interpreting IDR as a Petrov-Galerkin method, Report 09-10-22, Temple University, Oct., 2009. <http://www.math.temple.edu/~szyld/>
- [13] Sleijpen, G., Sonneveld, P. and van Gijzen, M.B., Bi-CGSTAB as an induced dimension reduction method, Applied Numerical Mathematics. (in print)
- [14] Sonneveld, P., AGS - IDR - CGS - BiCGSTAB - IDR(s): The circle closed, A case of serendipity, The Proc. of Int. Kyoto Forum 2008 on Krylov subspace methods, pp.1-14, September, 2008.
- [15] Sonneveld, P., van Gijzen, M.B., IDR(s): a family of simple and fast algorithms for solving large nonsymmetric linear systems, SIAM J. Sci. Comput., Vol. 31, No.2, pp.1035-1062, 2008.
- [16] 谷尾真明, 双共役勾配法の拡張に関する研究, 東京大学大学院 情報理工学系研究科修士論文, 2009.2.
- [17] 谷尾真明, 杉原正顯, 高次元の shadow residual を持つ Bi-CG 法に高次の加速多項式を付加したアルゴリズム, 日本応用数理学会 第 5 回研究部会連合発表会, 2009.3.
- [18] University of Florida Sparse Matrix Collection: <http://www.cise.ufl.edu/research/sparse/matrices/index.html>
- [19] van Gijzen, M.B., Sonneveld, P.: An elegant IDR(s) variant that efficiently exploits bi-orthogonality properties, TR 08-21, Math. Anal., Delft Univ. of Tech., 2008.
- [20] van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems, SIAM J. Sci. Comput., **13**, 2, pp.631-644, 1992.
- [21] Wesseling, P., Sonneveld, P., Numerical Experiments with a Multiple Grid- and a Preconditioned Lanczos Type Methods, Lecture Notes in Math., Springer, No.771, pp.543-562, 1980.
- [22] 張紹良, 藤野清次: ランチョスプロセスに基づく積型反復解法, 日本応用数理学会論文誌, **5**, pp.343-360, 1995.